



MICROTEK
IN-CIRCUIT EMULATORS

WindRiver Guide

Revision 1.0

15661-000

9/9/98

Table of Contents

PACKING LIST.....	1
INSTALLATION PROCEDURE FOR WINDRIVER DEVELOPER'S KIT	2
INVOKING THE INTERFACE.....	2
OUTSTANDING ISSUES AND KNOWN PROBLEMS.....	3
SUPPORTED EMULATION COMMANDS.....	4

Index of Supported Emulation Commands

ADACOMPILERUSED [NONE RATIONAL DDCI]	8
ADDDEFAULTEXTENSION [EXT1] [EXT2] ...	8
ADDRESSOF#MOD> <#MOD#FUNC> <#MOD#LINE> <#MOD#FUNC#VAR>	8
ICE ASM <STRING>	9
ICE ASMADDR [AUTO USE16 USE32] [<ADDRESS> [USER SMM IO]]	9
ICE BKPT [ENABLE DISABLE] [TEMPORARY PERMANENT] [<ADDRESS>] [@<ID>] [USER SMM]	10
ICE BKPTCLEAR @<ID> ALL <ADDRESS> [USER SMM]	10
BTM [ON OFF USER]	10
ICE BUSRETRY [ON OFF]	11
ICE CACHING [ENABLE DISABLE [DATA WRITEBACK]] <PENTIUM ONLY>	11
ICE CACHING [ENABLE DISABLE] <486 ONLY>	11
ICE CAUSE	11
ICE COMPORT [COM1 COM2 COM2 COM3 NETWORK NONE]	11
ICE CONFIG IGNOREHLDA [ON OFF]	11
ICE CONFIG EXTERNALBREAK [ACTIVEHIGH ACTIVELOW] <PENTIUM ONLY>	11
ICE CONFIG EMULATING [ACTIVEHIGH ACTIVELOW] <PENTIUM ONLY>	11
ICE CONFIGSYMBOLS [BASENAME]	12
ICE COPY <FROM_START_ADDRESS> <FROM_END_ADDRESS LENGTH <LEN>> [USER SMM IO] [TARGET] TO <TO_ADDRESS TARGET <TO_ADDRESS> TARGET> [USER SMM IO]	12
ICE DASM [AUTO USE16 USE32] [<START ADDRESS> [<END ADDRESS>] [USER SMM IO]]	12
ICE DASMSYM [ON OFF]	12
ICE DISABLEALARMLIMIT	12
ICE DISABLEHIGHWATERMARK	12
ICE DISPLAYSTACK [LOCALS HEX]	12
ICE DISPLAYSYMBOLS [MODULES FUNCTIONS PUBLICS LINES SORTED #<MODULE NAME>]	12
ICE DISPLAYTYPES [<TYPE ID>]	13
ICE DR [<0 1 2 3> [BKPT USER]]	13

ICE DR [<0 1 2 3> DATA <X W RW> <ADDRESS> <BYTE WORD LONG DWORD> [EXACT]]	13
ICE DR [<0 1 2 3> DATA <X W RW IO> <ADDRESS> <BYTE WORD LONG DWORD>] <PENTIUM ONLY>	13
ICE DRMODE [BKPT PINS] <PENTIUM ONLY>	13
ICE DT <SELECTOR_RANGE> <REG> [<BASE_ADDRESS>] [ALL <BASE_ADDRESS>: BASE <ADDRESS> [LIMIT <BYTES>]]	14
ICE DUMP [LOOP] <ADDRESS1> [<ADDRESS2>] [BYTE WORD LONG DWORD FLOAT DFLOAT LDFLOAT] [USER SMM IO]	14
ICE EMUSTATUS	14
ICE ENABLEALARMLIMIT	14
ICE ENABLEHIGHWATERMARK	14
ICE EVENTRESTORE <FILENAME>	14
ICE EVENTSAVE <FILENAME>	14
ICE FILL <ADDRESS1> <ADDRESS2> <DATA>... [BYTE WORD LONG DWORD] [USER SMM IO]	15
ICE FILLSTACKPATTERN	15
ICE FLASHERASE	15
ICE FLASHSET [<PART NUMBER> BYTE WORD [8 16 32 64]]	15
ICE FLUSH <486 AND P6 ONLY>	15
ICE FLUSH [DATA [NOWRITEBACK] [INSTRUCTION]] <PENTIUM ONLY>	15
ICE GDT [<SELECTOR_RANGE> <REG>] [<BASE_ADDRESS>] [ALL <BASE_ADDRESS>: BASE <ADDRESS> [LIMIT <BYTES>]]	16
ICE GETBASE [<BASENAME>]	16
ICE GO	16
ICE GOINTO [CALL RETURN] [LINE STATEMENT]	16
ICE GOUNTIL [CALL RETURN] [LINE STATEMENT]	16
ICE HALT	17
ICE HELP ["<TOPIC>"]	17
ICE HISTORY [<SIZE>]	17
ICE IDT [<INDEX_RANGE> <REG>] [<BASE_ADDRESS>] [ALL] [<BASE_ADDRESS>: BASE <ADDRESS> [LIMIT <BYTES>]]	17
ICE ISEMUHALTED (RETURNS 0 IF RUNNING, NON-ZERO IF HALTED)	17
ICE LDT [<SELECTOR_RANGE> <REG>] [<BASE_SELECTOR>] [ALL <BASE_SELECTOR>: SELECTOR <SELECTOR>]	17
ICE LOAD "<FILENAME>" [USER SMM] [CODE NOCODE] [SYMBOLS NOSYMBOLS] [DEMAND NODEMAND] [DEMANGLE NODEMANGLE] [UPDATEBASE NOUPDATEBASE] [LOADREGISTER NOLOADREGISTER] [CASE IGNORECASE] [WARN NOWARN] [STATUS NOSTATUS] [MODULE <MODULE PATH NAME>] [RELOAD]	18
ICE LOADSIZE [BYTE WORD LONG DWORD]	18
ICE MAP [CLEAR <ADDR BASE> [<END ADDR>] [TARGET FLASH] [RAM ROM ROMBRK NONE OFF] [<SPACE>]]	18

ICE MAP [CLEAR <ADDR BASE> [<END ADDR>] [FLASH]] <FLASH MAP ONLY>	18
ICE MAXBITFIELDSize [<16 32>]	18
ICE MAXMEMSIZE [8..16]	18
ICE MMXDISPLAY [<ADDRESS>] <MMX PROCESSORS ONLY>	18
ICE MODULEREFERENCE <MODULENAME> [<FILENAME>]	18
ICE MSR <MSR NUMBER> [[<VALUEH>] <VALUEL>] <P6 ONLY>	19
ICE NAMEOF <ADDRESS>	19
ICE NET ADDRESS [<IP ADDR> [<NETMASK> [<GATEWAY>]]]	19
ICE NET CONFIG [ADD DELETE DEFAULT <HOSTNAME>]	19
ICE PD [<ADDRESS RANGE>]	19
ICE PMODE	19
ICE PT <ADDRESS RANGE>	19
ICE RAMTST [LOOP] <ADDRESS1> <ADDRESS2> [USER SMM IO]	19
ICE REGISTER [<REGISTER NAME> [VALUE]]...	19
ICE REMOVESYMBOLS	19
ICE RESET [CPUONLY TARGET]	20
ICE RESETANDGO	20
ICE RESETLOADERS "[PATH TO THE LOADER.INI FILE]"	20
ICE RESTORECS "<FILENAME>"	20
ICE RESTOREMAP "<FILENAME>"	20
ICE RUNACCESS [ON OFF]	20
ICE SAVECS "<FILENAME>"	20
ICE SAVEMAP "<FILENAME>"	20
ICE SEARCH <START ADDRESS> <END ADDRESS> [NOT] <DATA>...	
[BYTE WORD LONG DWORD] [USER SMM IO]	20
ICE SETBASE <BASENAME> <ADDRESS>	21
ICE SETSTACKALARM <% OF STACK>	21
ICE SETSTACKAREA [<ADDRESS><SIZE> [FILLAREA]]	21
ICE SETSTACKBASE <ADDRESS>	21
ICE SETSTACKSIZE <STACK SIZE>	21
ICE SIGNAL [<SIGNAL NAME> [ENABLE DISABLE]]	21
ICE SIGNAL ALL ENABLE DISABLE	21
ICE SIZE [BYTE WORD LONG DWORD]	21
ICE STACKINFO	21
ICE STEP [INTO OVER] [<COUNT>]	22
ICE STEPSRC [INTO OVER] [LINE STATEMENT] [<COUNT>]	22
ICE SWATBASE [<BASE ADDRESS 0> [<BASE ADDRESS 1>]]	22
ICE SWATCMD START STOP CLEAR	22
ICE SWATGRAN [1 2 4 8 16 32 .. 256]	22
ICE SWATHEADER [[<FILENAME>] "HEADER STRING"]	22
ICE SWATMODE [OPA CA ECA]	22
ICE SWATRESTORESETUP "<FILENAME>"	23

ICE SWATRESULT <SYMBOL ADDRESS RANGE> [FUNCS LINES] [DETAIL]	23
ICE SWATRESULT <START ADDRESS> [<END ADDRESS>] [DETAIL]	23
ICE SWATRESULT TOTAL OTHER [DETAIL]	23
ICE SWATRESULTSTATUS	23
ICE SWATSAVE <FILENAME> [<LOAD FILENAME>] ADDR-RANGE1 [ADDR-RANGE2]	23
ICE SWATSAVE <FILENAME> [<LOAD FILENAME>] @<RANGE FILENAME>	23
ICE SWATSAVESETUP "<FILENAME>"	23
ICE SWATSTATUS	23
ICE SYMBOLCLOSEFILE	24
ICE SYMBOLOPENFILE <PATH AND FILENAME>	24
ICE TEST [LOOP] [REPEAT CONTINUE] [BRIEF VERBOSE] [EXCEPT] [<TEST NAME> <TEST NUMBER>...]	24
ICE TRCSAVE <FILENAME> [OVERWRITE] <INST BUS CLOCK> [<FRAME START> <FRAME END>] [ON OFF] [DELTA ABSOLUTE RELATIVE[<ZEROFRAME>]]	24
ICE TSS [<SELECTOR> <REG> <BASE_ADDRESS><TSS286 TSS386>] [ALL] <BASE_ADDRESS>: BASE <ADDRESS> [LIMIT <BYTES>]	24
ICE VARType <#VARNAME> [<TYPE ID> <TYPE NAME>[*]]	24
ICE VERIFY [ON OFF]	24
ICE VERSION	24
ICE WRITE [LOOP] <ADDRESS> <DATA>... [BYTE WORD LONG DWORD] [USER SMM IO]	25
ICE XLT <ADDRESS>	25

Packing List

1. CD containing the Microtek ICE Studio™ Software.
2. Manual

Installation Procedure for WindRiver Developer's Kit

Install the software as follows:

Step 1, Install Tornado 1.0.1 from the CD with the appropriate key.

Step 2, Install Microtek back end developer's kit from CD.

Step 3, Install the necessary firmware using the firmware install program located in program group: "Microtek ICE Studio".

Invoking the Interface

1. Launch the Tornado Registry. An MS-DOS window will appear and the cursor will be blinking with no prompt. Finally a new window will appear. The heading will say: "Tornado was unable to detect a RPC portmapper. Would you like to launch one now?"
2. Press the OK button on the window to start an RPC portmapper. After you press the OK key another window will appear and an RPC portmapper process will register itself to the TCP/IP stack.
3. Launch Tornado. It should come up with NO additional messages.
4. Invoke the target server by using the icon in the Tornado Program Group. When the backend initializes the first time it will request the user specify which serial port will be used. Enter the appropriate serial port by clicking the radio button next to it. Then a window will request you specify the baud rate. Use the radio button to specify the proper baud rate. After this the software will upload the state of the emulator and the connection is complete. When the target server begins you will see the following detail typed to the screen:

D:\Tornado\host\x86-win32\bin\tgtsvr.exe ([hostname@servername](#)): Tue Sep 15 16:00:31 1998

License request... authorized on Local Host.

Wind River Systems Target Server: NT/Win95 version

Attaching backend... succeeded.

Establishing Microtek connection... succeeded.

Attaching C++ interface... succeeded.

Attaching a.out OMF reader... succeeded.

5. Minimize the MS-DOS windows
6. Go to Tornado and under tools select the debugger. You are now connected to the emulator.
7. Use the command "ice reg" to see the processor's register values. If this works then the ice mechanism is installed correctly.
8. Use the gdb command "attach system" to connect to your target.

In order to exit the target server the user can either issue a Control-C or Control-Break.

Outstanding Issues and Known Problems

The following are issues that remain outstanding in the product.

1. Commands that create a large amount of output can cause the interface to hang. An example of this is using the dasm command with a specified range that exceeded 0x2A0 worth of output. So for instance if you specify "dasm 0x0 0x3FF" the interface will experience a general protection fault.
2. If you are using the debugger, after attaching you will have a default assembly window provided with the current disassembled instructions visible. While this disassembly window is open you cannot get the debugger to update or use the source file. You must first close the assembly window to get the source window to work.
3. If you are using a 386 or 486 emulator you will need to map some overlay memory to provide enough memory for the demo program to run on the SAST board. Using the ice command feature type "ICE MAP 0x0 0x3FFF", this specifies 256k of overlay for lower memory, enough for the demo program.

What are the main features of the PowerPack-Tornado integrated product?

- 1 The ITP or SW Plus or EA emulators provide a communication link between the target and host so that no target resources are required such as Ethernet or a serial communications channel.
- 2 With emulator control, the Tornado environment can be used to bring up Board Support Package (BSP) code as well as to debug hardware-software interfaces such as device drivers.
- 3 The emulator allows complete control of the processor to start and stop its execution. This has benefits over a target ROM monitor:
 - No target RAM or ROM resources are required, especially useful for limited memory designs.
 - A ROM/RAM monitor can be corrupted by errant programs that can write over the monitor state variables, causing the entire monitor to crash and require restarting. Since VxWorks does not support task-level hardware (MMU) protection, this is very important.
 - The emulator can always get control of the processor if it gets hung in a loop, or interrupts are disabled causing a comm port to no longer communicate with the CPU. More importantly, the emulator can show the user where in the program the processor got hung up so the bug can be fixed.
 - The Microtek emulator makes full use of the X86 on-chip debug registers to provide debugging through ROM code and the ability to set hardware breakpoints and variable recognition breakpoints that work even with caches enabled.

What are the basic feature-benefits of the integrated PowerPack-Tornado product?

The Tornado CrossWind debugger attaches to the target through the Microtek emulator, making available all the system-mode features of the debugger such as source-level stepping, setting and clearing breakpoints, viewing variables, plus starting and stopping the target processor. A summary of the PowerPack-Tornado features is:

Feature	Description	Benefit
Tornado "System Mode"	The VxWorks kernel is statically linked to the user application program and downloaded into the target. When the debugger stops the program, the processor is halted, therefore the target is under the complete control of the gdb/CrossWind for starting, stopping, and stepping the target processor.	Debug processor from usrConfig() function. Can debug initialization code that sets up hardware devices (including chipsets), device driver hardware, and in general, assists in debugging a BSP – Board Support Package.
gdb Shell	user can issue gdb commands and macros, plus tcl macros	Macros save time with repetitive operations. tcl macros can provide conditional breakpoints, checking target values, and other programmatic debugging activities.
"ice" command	The gdb "ice" command, added when the integrated product is installed, sends any SLD Shell command string to the emulator server for processing, thereby providing many of the emulator-specific commands to the user.	System-level and emulator-specific functions such as "reset" and "map" are made available with this command. Provides initialization of processor registers and/or I/O devices such as chipset programming. Provides Flash programming.
Target hardware connection	The emulator plugs into the processor socket or over top the processor chip. ITP products utilize the Intel Pentium or Pentium II Debug Port for access to the target.	User's target does not need to provide a serial or ethernet hardware channel to download and debug a target program

Tornado browser	Displays task-related information	Useful for viewing current state of all tasks, messages, and semaphores
WindSh	Tornado Shell	Can be used to send commands to target, although it has more limited features than the System mode.

What are the steps to get the CrossWind debugger up and running with the demonstration application?

The assumption is that Tornado has been installed on your PC and the appropriate license servers are running. This includes the (separate) target backend server that communicates to the Microtek emulator.

Start Tornado

Start up the target server

Start up CrossWind (Tools/Debugger...). Once it has started and (automatically) loaded gdb.tcl, issue the commands:

```
(gdb) attach system
```

This enables system mode and connects the debugger to the target, through the emulator. The assembly window opens at this point.

```
(gdb) loadvx
```

This tcl function was written specifically for the SAST demo program. It sets up the starting address of the program, loads the VxWorks kernel and demo software that has been linked in with the kernel, then it brings up the CrossWind source window. The generic load program is `loadf("filename")`.

Now standard source-level debugging can begin. You can set breakpoints in the application code, viewed from the source window. Variables can be browsed, registers viewed from the register window, and memory viewed from the memory window.

Supported Emulation Commands

Command syntax conventions

Notation	Meaning
CommandName	Case is insignificant in command names or aliases.
<placeholder>	Substitute value or symbol arguments for placeholders.
[option]	Repeat bracketed items no more than once. Brackets are not part of the command, unless otherwise noted.
{<many_values>}	Repeat items in braces zero or more times. Braces are not part of the command, unless otherwise noted.
<series>...	An ellipsis indicates a series of repeating items.
Option_1 Option_2	Select one of the options separated by vertical lines.
"<string_constant>"	Surround string constants by double quotation marks.
/* comment */	Delimit comments C-style.
//command output	Command output is preceded by forward slashes.
<address>	Use a linear, physical, virtual, or symbolic address.

Following is the list of commands you will see if you type the "ice help" command. Although they are all included, not all are functional in the Tornado implementation. For instance all of the commands used to modify, and display symbols do not work. The reason is all of the symbol information remains in the Tornado interface, so none make it to the symbol table of the emulator. Also commands that are used to specifically affect variables used by the native interface will have no effect. This includes commands like DisableHighWaterMark etc. These commands are noted in the commands description.

In general commands that return information about the status of the emulator and target, like "ice pmode" for processor mode, or "ice dasm" for disassembling memory are safe to use in all situations. Commands that affect the state of the target, like "ice go" or "ice halt" or "ice bkpt" can leave the system in a state where the gdb interface is not "current". This normally does not cause a problem and will "sync" if a control command is issued through the gdb interface like "step". It is recommended that ice control commands be used sparingly.

For more complete explanations of these commands please refer to the SLD help files and on-line manuals.

adaCompilerUsed [None|Rational|DDCI]

Not applicable to the Tornado implementation

addDefaultExtension [ext1] [ext2] ...

Not applicable to the Tornado implementation

addressOf#mod>|<#mod#func>|<#mod#line>| <#mod#func#var>

Used to locate a variable address in standard SLD. Not applicable to the Tornado implementation.

```
ice asm <string>
```

<string> is an Intel assembly language statement.

Check the syntax of <string> and write the instruction bytes to memory at the current assembly address. (Set the current assembly address with AsmAddr.)

Symbolic assembly not supported.

Example:

```
ice "asm jmp 400"
```

Will put the Intel opcode for an unconditional jump into memory at the current memory location.

```
ice asmAddr [auto|use16|use32] [<address> [user|smm|io]]
```

Sets the address where assembly instructions will be inserted.

Auto - derives the addressing mode based on the pmode

Use16 - uses 16-bit operands and addresses.

Use32 - uses 32-bit operands and addresses.

<address> is a numeric or symbolic address of the location where the next ASM command will write.

<space> - specifies the Intel address space as user, smm, or io.

With no address, AsmAddr displays the current assembly address.

Example:

```
ice "asmaddr 0x1000"
```

Will move the position where new opcodes will be added to memory to address location 0x1000. The next ice "asm mov ax,100" will be put in memory address 0x1000.

```
ice bkpt [enable|disable] [temporary|permanent] [<address>] [@<id>]
[user|smm]
```

Display, set, or modify breakpoints.

Example:

```
ice "bkpt 0x153FFF"
```

sets a breakpoint at the specified memory location 0x153FFF.

```
ice "bkpt"
```

```
@0 0x153FFF
```

Will print out the current breakpoints that are set. In this case we have set one breakpoint.

```
ice "bkpt disable @0"
```

This command will disable the breakpoint 0.

```
ice "bkpt 0x3000L smm"
```

Will set a breakpoint in system management memory at memory location 0x3000.

```
ice bkptClear @<id>|all|<address> [user|smm]
```

Remove Breakpoints. Use ice bkptClear to remove a specified breakpoint or all temporary and permanent breakpoints.

Example:

```
ice "bkptClear @0"
```

This command will remove the breakpoint specified in list place 0.

```
BTM [on|off|user]
```

A command used to control the processor's use of branch taken messages. When BTM's are ON, the emulator is capable to tracing though cache and telling the difference between fetched and executed instructions. It is also necessary in code coverage.

ice busRetry [on|off]

Asserts bus error after timeout.

ice caching [enable | disable [data | writeback]] <Pentium only>

Used to enable and disable caching on a Pentium emulator. It also allows the user to specifically enable data or writeback features of the Pentium processor.

ice caching [enable | disable] <486 only>

Used to enable/disable caching on a 486.

ice cause

Use this command when the emulator is halted to discover the reason for the most recent halt.

ice comPort [com1|com2|com2|com3|network|none]

Used to setup the communication port for the emulator. Gives you the choice of working with serial ports, or a network.

ice config ignoreHLDA [on|off]

Config ignoreHLDA on causes the emulator to ignore the HLDA pin state. Set config ignoreHLDA on when HLDA is programmed as an I/O bit.

Config IgnoreHLDA off causes the emulator to examine the HLDA pin state before generating overlay RAM or trace/trigger strobe.

ice config ExternalBreak [activeHigh|activeLow] <Pentium only>

Used to configure the external input signals to be active either high or low. This is target dependent.

ice config Emulating [activeHigh|activeLow] <Pentium only>

```
ice configSymbols [basename]
```

Configsymbols is used to update the symbols with the base address obtained from the descriptor table. This is not useful in the Tornado implementation.

```
ice copy <from_start_address> <from_end_address | LENGTH <len>>  
[user|smm|io] [TARGET] TO <to_address | TARGET | <to_address>  
TARGET> [user|smm|io]
```

Copies from one region of target or overlay memory to another.

```
ice dasm [auto|use16|use32] [<start address> [<end address>]  
[user|smm|io]]
```

Disassemble Intel opcodes in memory.

```
ice dasmSym [on|off]
```

Turns on and off symbolic disassembly. Not useful in the tornado implementation.

```
ice disableAlarmLimit
```

Disable the warning message for excessive stack useage. Not applicable to the Tornado implementation.

```
ice disableHighWaterMark
```

Disables keeping trace of stack usage. Not applicable to the Tornado implementation.

```
ice displayStack [locals|hex]
```

Displays the stack frames. Not applicable to the Tornado implementation.

```
ice displaySymbols [modules|functions|publics|lines|sorted|/#<module  
name>]
```

Displays symbols. Not applicable to the Tornado implementation.

```
ice displayTypes [<type id>]
```

Displays the current types used in a file. Not applicable to the Tornado implementation.

```
ice dr [<0|1|2|3> [bkpt|user]]
```

Allows the user to specify if debug registers will be used by the debugger or another program.

```
ice dr [<0|1|2|3> data <x|w|rw> <address> <byte|word|long|dword> [exact]]
```

Allows the user to setup a breakpoint with a data value and an address.

Data - configures the register as a data read/write breakpoint.

<mode> (x/w/rw) - specifies if the breakpoint will activate if the address is executed, written, or read/write.

<address> specifies the virtual or linear base address of the breakpoint.

<size> - specifies the 1,2, or 4 bytes starting with <address> as the address range of the data breakpoint. Emulation breaks on any data access in this range.

```
ice dr [<0|1|2|3> data <x|w|rw|io> <address> <byte|word|long|dword>] <Pentium only>
```

Allows the user to specify i/o addresses with same functionality supported as the dr command above. Can be used with Pentium targets only.

```
ice drMode [bkpt | pins] <Pentium only>
```

Allows the user to configure the debug registers as breakpoints or to activate the breakpoint pins of the processor.

```
ice dt <selector_range> | <reg> [<base_address>] [all  
<base_address>: base <address> [limit <bytes>]
```

Used to display the descriptor table. Use dt to display the descriptor tables entries for a single selector or range of selectors. The selector displayed is determined by <selector range>, <register>, or base <address>, one of which must be specified.

```
ice dump [loop] <address1> [<address2>]  
[byte|word|long|dword|float|dfloat|ldfloat] [user|smm|io]
```

Used to dump memory contents. NOTE: excessive size will be truncated in the Tornado implementation.

```
ice emuStatus
```

Used to display the current status of the emulator.

```
ice enableAlarmLimit
```

Used to control warnings on the native interface. Not applicable in the Tornado implementation.

```
ice enableHighWaterMark
```

Used to control warnings in the native interface. Not applicable in the Tornado implementation.

```
ice eventRestore <filename>
```

Used to read stored event definitions used in the triggering system. Not applicable to the Tornado implementation.

```
ice eventSave <filename>
```

Used to save event definitions used in the triggering system. Not applicable to the Tornado implementation.

```
ice fill <address1> <address2> <data>... [byte|word|long|dword]
[user|smm|io]
```

Used to fill memory with data.

<address1> is the first address in the region to be filled.

<address2> is the last address in the region to be filled.

data - is the data to be written.

byte - specifies the data is a byte value.

word - specifies the data is a word value

long, dword - specifies the data is a double word value.

<space> - Specifies smm or user (default) address space

```
ice fillStackPattern
```

Used to specify the pattern used to fill stack space. Not applicable to the Tornado implementation.

```
ice flashErase
```

Command used to erase a section of flash memory specified in a target.

```
ice flashSet [<part number> byte|word [8|16|32|64]]
```

Used to set the size of memory access to a flash area. The Intel or AMD flash partnumber is used along with the width of the desired access.

```
ice flush <486 and P6 only>
```

Used to flush the cache of the processor.

```
ice flush [data [nowriteback] | [instruction]] <Pentium only>
```

Used to flush the cache of a Pentium processor.

```
ice gdt [<selector_range>|<reg>] [<base_address>] [all]
<base_address>: base <address> [limit <bytes>]
```

Used to print out the gdt record values. The global descriptor table entries for a single selector or range of selectors can be specified. With no parameters gdt show all valid entries in the range gdt_base to gdt_base+gdt_limit.

```
ice getBase [<basename>]
```

Used to output base names and their respective base addresses. When used with the <basename> parameter, it outputs only the address value of that basename. When used without a basename parameter, it outputs all base addresses. Not applicable to the Tornado implementation.

```
ice go
```

Used to make the emulator begin emulation. NOTE: when using this and other ice commands that change the location of the program counter or other processor resources, the Tornado interface may not be aware of the change causes synchronization issues. Best reserved for use before gdb is "attached" to the target system.

```
ice goInto [call|return] [line|statement]
```

Used to direct a single step into a called routine. . NOTE: when using this and other ice commands that change the location of the program counter or other processor resources, the Tornado interface may not be aware of the change causes synchronization issues. Best reserved for use before gdb is "attached" to the target system.

```
ice goUntil [call|return] [line|statement]
```

Used to direct the emulator to execute until it reaches a call or a return. . NOTE: when using this and other ice commands that change the location of the program counter or other processor resources, the Tornado interface may not be aware of the change causes synchronization issues. Best reserved for use before gdb is "attached" to the target system.

ice halt

Used to halt emulation. . NOTE: when using this and other ice commands that change the location of the program counter or other processor resources, the Tornado interface may not be aware of the change causes synchronization issues. Best reserved for use before gdb is "attached" to the target system.

ice help ["<topic>"]

Used to print out the available commands.

ice history [<size>]

Used to set the size of the command history. Not applicable to the Tornado implementation.

ice idt [<index_range>|<reg>] [<base_address>] [all] [<base_address>: base <address> [limit <bytes>]]

Used to display the interrupt descriptor table.

<index range> specifies either a single value or two values to specify a range.

<register> specifies a register, the selector for the specified register is used.

base <address> specifies the descriptor table base address.

[limit <bytes>] If a base address is specified, you must also specify either <selector_range> or limit <bytes> to define the range to be displayed.

all - displays all of the entries, including invalid or reserved.

ice isEmuHalted (returns 0 if running, non-zero if halted)

Used to determine if the emulator is running or not.

ice ldt [<selector_range>|<reg>] [<base_selector>] [all] <base_selector>: selector <selector>

Used to output the local descriptor table records.

```
ice load "<filename>" [user|smm] [code|nocode] [symbols|nosymbols]
[demand|nodemand] [demangle|nodemangle]
[updatebase|nouupdatebase] [loadregister|noloadregister]
[case|ignorecase] [warn|nowarn] [status|nostatus] [module <module
path name>] [reload]
```

Used to load omf images. Not applicable to the Tornado implementation.

```
ice loadSize [byte|word|long|dword]
```

Used to set the memory write-access size during a file load. Not applicable to the Tornado implementation.

```
ice map [clear|<addr base> [<end addr>] [target|flash]
[ram|rom|rombrk|none|off] [<space>]]
```

Used to control the overlay map.

```
ice map [clear|<addr base> [<end addr>] [flash]] <FLASH MAP
ONLY>
```

Used to control mapping of flash downloads.

```
ice maxBitFieldsiZe [<16|32>]
```

```
ice maxMemSize [8..16]
```

Used to set the maximum bit field size for OMF386 loadfiles. Not applicable to the Tornado implementation.

```
ice mmxdisplay [<address>] <MMX processors only>
```

Used to display the contents of the MMX registers.

```
ice moduleReference <modulename> [<filename>]
```

Used to locate references to a specific module name. Not applicable to the Tornado implementation.


```
ice msr <msr number> [[<valueh>] <valuei>] <P6 only>
```

Used to output the values of the MSR registers.

```
ice nameOf <address>
```

Used to locate the symbol name for a specific address. Not applicable to the Tornado implementation.

```
ice net address [<IP addr> [<netmask> [<gateway>]]]
```

Used to set the net address of the emulator.

```
ice net config [add|delete|default <hostname>]
```

Used to manage the names of various emulators on a network.

```
ice pd [<address range>]
```

Used to display the page directory.

```
ice pmode
```

Used to display the current mode of the processor.

```
ice pt <address range>
```

```
ice ramtst [loop] <address1> <address2> [user|smm|io]
```

Used to run a par 3 ram test over the range specified.

```
ice register [<register name> [value]]...
```

Used to display or modify the value of a register. Default with no register specified outputs all registers.

```
ice removeSymbols
```

Used to clear the symbol table. Not applicable on the Tornado implementation.

```
ice reset [cpuonly|target]
```

Used to reset the target processor.

```
ice resetAndGo
```

Used to reset the processor and immediately begin emulating.

```
ice resetLoaders "[path to the LOADER.INI file]"
```

Used to control the loaders. Not applicable to the Tornado implementation.

```
ice restoreCS "<filename>"
```

Used to restore Chip Selects. Not applicable to the Tornado implementation.

```
ice restoreMap "<filename>"
```

Used to restore a saved map file. Not applicable to the Tornado implementation.

```
ice runAccess [on|off]
```

Used to enable and disable the ability to read variable results while running. Not applicable to the Tornado implementation.

```
ice saveCS "<filename>"
```

Used to save chip selects. Not applicable to the Tornado implementation.

```
ice saveMap "<filename>"
```

Used to save an overlay map to a file. Not applicable to the Tornado implementation.

```
ice search <start address> <end address> [not] <data>...  
[byte|word|long|dword] [user|smm|io]
```

Used to search through memory for a particular pattern.

ice setBase <basename> <address>

Used to set the base address of a particular base. Not applicable to the Tornado implementation.

ice setStackAlarm <% of stack>

Used to control the stack warning mechanism. Not applicable to the Tornado implementation.

ice setStackArea [<address><size> [fillArea]]

Used to redefine the stack area. Not applicable to the Tornado implementation.

ice setStackBase <address>

Used to redefine the stack base. Not applicable to the Tornado implementation.

ice setStackSize <stack size>

Used to set the size of the stack. Not applicable to the Tornado implementation.

ice signal [<signal name> [enable|disable]]

ice signal all enable|disable

Used to display the settings for signals. Also used to enable/ disable signals.

ice size [byte|word|long|dword]

Used to set the memory access size for the target.

ice stackInfo

Used to display stack information. Not applicable to the Tornado implementation.

ice step [into|over] [<count>]

Used to step the processor through an opcode. Into will execute into calls, over will specify to count a call as one instruction, count specifies how many steps.

ice stepSrc [into|over] [line|statement] [<count>]

Used to step emulator by source lines. Not applicable to the Tornado implementation.

ice swatBase [<base address 0> [<base address 1>]]

Used to set the memory location where swat measurements will be taken. For example, if you wish to check the code coverage of a particular set of functions, you would set the swat base to an address at the beginning of these functions in memory.

ice swatCmd start|stop|clear

Used to start the swat gathering information, stop the swat gathering of information, and the clearing of information.

ice swatGran [1|2|4|8|16|32|..|256]

Used to set the granularity of the swat measurements. If you wish each byte to be checked for coverage then use 1, if each word is appropriate, use 2, etc..

ice SwatHeader [[<filename>] "Header String"]

Allows the user to specify the header string for the swat report.

ice swatMode [opa|ca|eca]

Sets the type of measurement the swat will perform. OPA = Overall Performance Analysis, CA = Code Coverage Analysis, Enhanced Code Coverage.

ice SwatRestoreSetup "<filename>"

Used to retrieve a previously saved setup for swat.

ice swatResult <symbol address range> [funcs|lines] [detail]

Not applicable to Tornado implementation

ice swatResult <start address> [<end address>] [detail]

Not applicable to Tornado implementation

ice swatResult total|other [detail]

Not applicable to Tornado implementation

ice swatResultStatus

Not applicable to Tornado implementation

ice SwatSave <filename> [<load filename>] addr-range1 [addr-range2]

Not applicable to Tornado implementation

ice SwatSave <filename> [<load filename>] @<range filename>

Used to save the results of a particular analysis

Not applicable to Tornado implementation

ice SwatSaveSetup "<filename>"

Used to save a setup for swat for later use.

ice swatStatus

Not applicable to Tornado implementation

ice symbolCloseFile

Used to control reading from a load file. Not applicable to the Tornado implementation.

ice symbolOpenFile <path and filename>

Used to control ready from a load file. Not applicable to the Tornado implementation.

ice test [loop] [repeat|continue] [brief|verbose] [except] [<test name>|<test number>...]

Used to run confidence test repeatedly.

ice trcsave <filename> [overwrite] <inst | bus | clock> [<frame start> <frame end>] [on | off] [delta | absolute | relative[<zeroframe>]]

Used to save trace to a file. Not applicable to the Tornado implementation.

ice tss [<selector>|<reg>|<base_address><tss286|tss386>] [all] <base_address>: base <address> [limit <bytes>]

Used to display task state selectors.

ice varType <#varname> [<type id>|<type name>[*]]

Used to specify what variable type a symbol has. Not applicable to the Tornado implementation.

ice verify [on|off]

Used to specify that all accesses to target memory be checked. Especially useful in downloads to insure they are actually getting stored validly.

ice version

Used to output the current version of the software.

```
ice write [loop] <address> <data>... [byte|word|long|dword]  
[user|smm|io]
```

Used to write to an area of memory.

```
ice xlt <address>
```

Used to translate an Intel code segment: offset address into a linear address.